

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/24508 A2

(51) International Patent Classification⁷: **H04N**

(74) Agent: JEREMY M. BEN-DAVID & CO. LTD.; Har
Hotzvim Hi-Tech Park, P.O. Box 45087, 91450 Jerusalem
(IL).

(21) International Application Number: **PCT/IL00/00607**

(22) International Filing Date:
28 September 2000 (28.09.2000)

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,
DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,
NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,
TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
132144 30 September 1999 (30.09.1999) **IL**

(84) Designated States (*regional*): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG,
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

(71) Applicant (*for all designated States except US*): **WEB-
SPLIT.COM INC.** [US/US]; 1209 Orange Street, County
of Newcastle, Wilmington, DE 19801 (US).

(72) Inventor; and

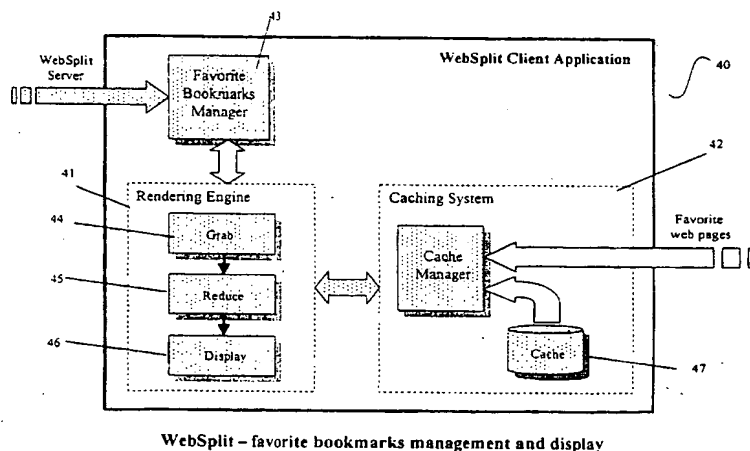
(75) Inventor/Applicant (*for US only*): **HAROUSH, Ariel**
[IL/IL]; 18 Habaron Hirsch Street, 96148 Jerusalem (IL).

Published:

— Without international search report and to be republished
upon receipt of that report.

[Continued on next page]

(54) Title: A METHOD OF FORMING NETWORK USER PROFILES



(57) Abstract: Disclosed is a facile method of forming a dynamic psycho-ergonomic profile of users, for correlation of like users, and accordingly for forming common profile "societies"; substantially including the steps of: (A) forming a repository for personal bookmark lists for a plurality of users; (B) maintaining respective personal bookmark list in the repository; (C) logically aggregating a collective list of all bookmarks maintained in the repository; (D) logically structuring a combinatorial set of n-tuplings of bookmarks in the collective list; and (E) providing substantially each user of the plurality of user with read-only access to the logical links of any n-tupling wherein said each user is likewise logically linked. Furthermore, disclosed is a rendering engine, tangibly embodying a program of instructions executable by the machine to perform method steps for presenting split screen bookmark imagery in conjunction with a fetch and convert engine and with the repository, the method including the steps of creating an intermediate image to hold horizontal zoom; pre-calculating filter contributions for a row; and placing each calculated pixel into a common intermediate image.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A METHOD OF FORMING NETWORK USER PROFILES

FIELD OF THE INVENTION

The present invention generally relates to a method for similar Internet related information affiliation clustering, such as for finding and representing groups of common-interest Internet users.

More specifically, the present invention relates to forming a search space that is oriented according to profiles of personal or other predetermined interests, and to dynamic rendering methods used therewith.

BACKGROUND OF THE INVENTION

Dynamic communications networks are groups of common interest users who communicate with each other. Often such networks are formed by asking users to register. This may facilitate a mechanism for organizing a newsletter where users may contribute their respective ideas or for routing intercommunications in a working group. Such networking forums are implemented using virtually any technology. For example, a mailing list sent by national post, a private corporate telephone exchange, an Intranet, or the likes, all form the requisite infrastructure for a well-defined intercommunicating group.

Today, with the Internet as a common communications infrastructure, there exist the possibility of creating common interest groups according to user profile or user preferences or other predefined criteria. Therefore, instead of a bulletin board facilitating respectively political action, dating, e-commerce, or group event registration, it is now possible to create groups whose common interest can only be statistically defined. In today's society, where a person's interpersonal contacts are often dominated by his job function or by his geographic location, but not by his personal interests, there is a long-felt need for a communications infrastructure wherein common personal interests define groups. More specifically, there is a need

in the art for a facile method of forming a dynamic psycho-ergonomic profile of users, which may be used for correlation of like users and accordingly for forming common profile "societies". Furthermore, there is a need in the art for methods of rendering concurrent graphic-contents, such as those representing members of such societies or their mutual plurality of interests.

NOTICES

Numbers, alphabetic characters, and roman symbols are designated in the following sections for convenience of explanations only, and should by no means be regarded as imposing particular order on any method steps.

Likewise, the present invention will forthwith be described with a certain degree of particularity, however those versed in the art will readily appreciate that various modifications and alterations may be carried out without departing from either the spirit or scope, as hereinafter claimed.

GLOSSARY

"WebSplit" relates to a representation of personal or common interest profiled user groups; the software and database support requisite for forming and maintaining such groups, and presumably to groups formed thereby. Furthermore, WebSplit relates to a resultant graphic rendering of substantially common interest; real or potential.

An "n-tuple" relates to a group (from the word "multiple"). If the group has one principle item, then the group is a 1-tuple; having two principle items, it's a 2-tuple; having three principle items, it's a 3-tuple; etc.

A "combinatorial set of n-tuplings" is an arbitrary selection and ordering of n-tuples. For example, a combinatorial set of n-tuplings may be all of the 1-tuples.

2-tuples, and 3-tuples of all possible tuples, which are logically grouped together and then sorted according to a concatenation of principle items respectively therein.

A "repository" is a database, a data structure, a collection of electronic files, a combination of any of the aforesaid, or the like.

A "bookmark" is an address on a network, such as on the Internet or on an Intranet, to a site or a WWW-page or a file or the like. Generally, a bookmark may also relate to any web application for managing graphics associated contents including homepages application, video-conference application, interactive photo album (on the web) application, e commerce application - browsing to display many goods or products on WebSplit's frames, a "one to one" shop - a custom made shop who tailored on one's particular profile and needs, and displays products on WebSplit's frames, an application who runs few other applications and displays their outputs in each of WebSplit's frames, or even a generic application that incorporate some or all of the above under a common shell; or to descriptors, links, or pointers related to any of the aforesaid.

SUMMARY OF THE INVENTION

The present invention relates to a facile method of forming a dynamic psycho-ergonomic profile of users, for correlation of like users, and accordingly for forming common profile "societies".

This method includes the steps of:

- a) **forming** a repository for personal bookmark lists wherein the repository is accessible to at least one network server;
- b) each user of a plurality of users **maintaining** his respective personal bookmark list in the repository;
- c) logically **aggregating** a collective list of all bookmarks maintained in the repository wherein
 - i) each bookmark is listed only once, and

- ii) each user maintaining that respective bookmark is listed in association therewith;
- d) logically **structuring** a combinatorial set of n-tuplings of bookmarks in the collective list wherein for each n-tupling there are logical links with each respective user maintaining every bookmark in the n-tupling; and
- e) **providing** substantially each user of the plurality of user with read-only access to the logical links of any n-tupling wherein said each user is likewise logically linked.

Simply stated, the present invention presumes that an especially significant factor in characterizing a person's interests is the bookmarks to Internet sites that the person maintains and uses; and perhaps other applications to which he subscribes. (Note: Such applications are substantially bookmark equivalent - in the context of the present invention.) If a person reports this maintenance and use to a central location, for example to a WWW accessible server, then the central location can construct a psycho-ergonomic profile of the user.

Using his own profile, this user can discover other users having similar profiles; provided such "other user users" likewise participate in this reporting process, and providing that these profiles (or correlation sets including them) are also accessible. Likewise, this collection of maintained profiles provides a valuable resource to market researchers, sociologists, communications scientists, etc.

Furthermore, the present invention relates to a rendering engine, tangibly embodying a program of instructions executable by the machine to perform method steps for presenting split screen bookmark imagery in conjunction with a fetch and convert engine and with a repository, the method including the steps of **creating** an intermediate image to hold horizontal zoom; **pre-calculating** filter contributions for a row; and **placing** each calculated pixel into a common intermediate image.

On the one hand, the rendering engine of the present invention is a facile tool for ergonomically representing results from the present facile method of forming a dynamic psycho-ergonomic profile of users. On the other hand, the rendering

engine of the present invention is a facile tool for ergonomically representing results for simulated common interest groups.

The present invention further relates to software systems including a tuple of system components selected from the list: a rendering engine, a fetch and convert engine, and a repository.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to understand the invention and to see how it may be carried out in practice, embodiments including the preferred embodiment will now be described, by way of non-limiting example only, with reference to the accompanying drawings, in which:

Figure 1 illustrates a schematic diagram of the present method;

Figure 2 illustrates a schematic diagram of a program storage device embodying instructions for executing the present method;

Figures 3-6 illustrates schematic diagrams of organizational aspects of the preferred embodiment of the present method; and

Figure 7 illustrates a schematic diagram of organizational aspects of main components enabling simultaneous bookmark representational viewing.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a facile method of forming a dynamic psycho-ergonomic profile of users, for correlation of like users, and accordingly for forming common profile "societies", the method including (see figure 1) the steps **10** of:

- a) **forming 11** a repository for personal bookmark lists wherein the repository is accessible to at least one network server;
- b) each user of a plurality of users **maintaining 12** his respective personal bookmark list in the repository;

- c) logically **aggregating 13** a collective list of all bookmarks maintained in the repository wherein
 - i) each bookmark is listed only once, and
 - ii) each user maintaining that respective bookmark is listed in association therewith;
- d) logically **structuring 14** a combinatorial set of n-tuplings of bookmarks in the collective list wherein for each n-tupling there are logical links with each respective user maintaining every bookmark in the n-tupling; and
- e) **providing 15** substantially each user of the plurality of user with read-only access to the logical links of any n-tupling wherein said each user is likewise logically linked.

According to an embodiment of the present invention, the step of **providing** includes installing a rendering engine on a terminal associated process of a user of any user of the plurality of users, wherein the rendering engine maps content associated with the logical links of read-only access onto the associated terminal.

According to another embodiment of the present invention, the step of **maintaining** includes **accepting** personal data about the respective user. There are two variations of this embodiment, which are mutually compatible.

According to a first variation on this embodiment of the present invention, the step of **accepting** includes **querying** the respective user for his personal data, **receiving** his response to the query, and **associating** a stored representation of the response with his respective personal bookmark list. Furthermore, according to an extension of this variation, the step of **aggregating** includes **characterizing** the stored representation of a respective user's personal data response and **maintaining** this characterization as at least one pseudo-bookmark in the user's respective personal bookmark list, wherein pseudo-bookmarks are aggregated into the collective list substantially like other maintained bookmarks.

According to a second variation of this embodiment, the step of **accepting** includes **querying** a database for personal data associated with the respective user.

receiving a response to the query, and **associating** a stored representation of the response with the user's respective personal bookmark list. Furthermore, according to an extension of this variation, the step of **aggregating** includes **characterizing** the stored representation of a respective user's data-base provided personal data and **maintaining** this characterization as at least one pseudo-bookmark in the user's respective personal bookmark list, wherein pseudo-bookmarks are aggregated into the collective list substantially like other maintained bookmarks.

The present invention also relates to (see figure 2) a program storage device 21 readable by machine 22, tangibly embodying a program of instructions 23 executable by the machine to perform method steps for forming a dynamic psycho-ergonomic profile of users, said method steps including:

- a) **forming** a repository for personal bookmark lists wherein the repository is accessible to at least one network server 24;
- b) each user of a plurality of users **maintaining** his respective personal bookmark list in the repository;
- c) logically **aggregating** a collective list of all bookmarks maintained in the repository wherein
 - i) each bookmark is listed only once, and
 - ii) each user maintaining that respective bookmark is listed in association therewith;
- d) logically **structuring** a combinatorial set of n-tuplings of bookmarks in the collective list wherein for each n-tupling there are logical links with each respective user maintaining every bookmark in the n-tupling; and
- a) **providing** substantially each user of the plurality of user with read-only access to the logical links of any n-tupling wherein said each user is likewise logically linked.

WebSplit – a general overview

The WebSplit project brings the split screen of the TV, or the channel-1 cable TV to the internet. The idea behind the project is to let the Internet surfers to view

many web pages on a single split screen application, to have their content updated at the moment the web page is modified on the web server without forcing the user to manually refresh it, and to use various tools to meet other users who share some common interest and associated with each other – base on their private preferences, regarding to their favorite bookmarks selection.

Turning now to Figures 3-6, wherein are illustrated a schematic diagrams of organizational aspects of the preferred embodiment of the present method; including a client server model **30**, a client application **40** in for favorite bookmark management and display in the context of the model, a server architecture **50** in the context of the model, and a communications model architecture **60** in the context of the model.

WebSplit desktop client application

WebSplit client application is linked to WebSplit server and design to serve the following purposes:

To allow the user to view his preferred favorite bookmarks on a split screen window, in order to see as many web pages and information as he wish, at a single glance.

To allow the user to manage and display his own bookmarks and private links on remote servers and to access them from every part of the world.

To allow the user to have an updated visual representation of his bookmarked web pages at any given time, and to be alerted when a certain page has been changed.

To present an updated statistical information about “gathering” around web sites the user is interested in.

To allow connecting people who share a common ground of interests, and are either watching the same content, or interested in the same web sites or pages (by bookmarking them as favorite sites); provide them with services like: instant messaging, chatting, searching tools, personal matching agents, etc.

To expose the users to personalized advertisement that will be pushed to him according to his personal profile. His personal profile, among other things will be constructed according to keywords in meta tags founded inside web pages he specified as his favorite bookmarks.

WebSplit servers

WebSplit server is basically designed to allow WebSplit clients to access the database in order to receive and store necessary information, such as: favorite bookmarks, personal details, etc. Among other things it allow and manage the communication among WebSplit users, runs management processes to have all registered web sites and pages ranked according to their popularity at any given time, produce rating numbers, etc.

The client/server model

WebSplit project takes the traditional client server model **30** to the internet by letting the WebSplit desktop client **31** application communicating with WebSplit server **32** over the internet **33**. The communication model between the clients and VuMark servers, along with a dedicated in-house developed communication infrastructure, supports the handling of the multi-sessions nature of the product.

The WebSplit client maintains communication sessions with the server in order to receive and store user personal data, and user personal favorite bookmarks. Also, the communication sessions allows the client to receive rating numbers of favorite domains or web pages, let users of WebSplit to search for and communicate with each other, etc.

WebSplit – a detailed description

The following sub-section will deal mostly with the internal parts of WebSplit protect. Most of it is about architecture of the software components, while other parts will discuss the functionality and the implementation.

WebSplit desktop client application

As described above, the main feature of the WebSplit client application is the split screen. The whole idea of the split screen is to deliver the feeling of a TV look alike split screen of channel-1 of the cables network, that allows the user to have a view of multiple web pages (channels) in a single glance. Therefore, the split screen occupies the most of the area of the WebSplit client window.

Other parts of the WebSplit client are the rendering engine, the caching system, and components for managing favorite bookmarks, communities of users with common interests, searching tools, etc.

Favorite bookmarks manager

One of the key components of WebSplit client application is the favorite bookmarks manager 40. It allows the user to manage his own private collection of bookmarks, while handling basic operations such as adding, modifying and removing of private web sites addresses, as well as importing other favorite bookmark files from the two popular browsers of Microsoft and Netscape. IE and Navigator, respectively.

The WebSplit client application provides the user with an editing tool to organize his bookmarks collection. Arranging the bookmarks list is done on a tree view who represents both the bookmarks list with the hierarchy of folders and URL's. The user performs editing operations on the tree and is able to save it to the WebSplit database, where it is kept for him.

Rendering engine

WebSplit rendering engine 41 is a collection of software components who are responsible for bringing small representation of (possibly many) web pages to WebSplit desktop client application. In order to carry out this task, the rendering engine incorporates three phases: grabbing, reducing and displaying.

Grabbing phase

In the grabbing phase **44** we capture the “image” (or the bitmap representation) of a web page - at the moment we render it onto a “virtual” window of a standard web browser.

In every windowing system, when an application displays its output on the screen by rendering the internal data structures to a window – the application memory holds the “raster” image of the output, before it is about to be clipped and mapped to the target window. At that moment we grab the image of the web page.

Since we only want to capture the raster image representation of the web page, we treat this standard web browser window as an “intermediate” window that helps us only with turning the web page to a raster image. So, there is no need for us to “realize”, or create a “visible” instance of that “intermediate” window – since we need it only for that purpose. It is there only to allow us grabbing the image, and then to pass it on to the next phase, the “reducing phase”.

Reducing phase

The actual reduction procedure **45** is an essential part of the rendering engine. As already described above, since we want to represent many (as many as the user will define) web pages on a split screen - we need to reduce the “image” of a web page. A reduced web page is a scaled-down representation of it: bookmarked web pages are displayed smaller than their original dimensions, in order to let more information to be exposed at once.

There are two methods for reducing web pages to a smaller (or even miniature) representation: static reduction and dynamic reduction. The WebSplit client employs both techniques on different situations when different level of interactivity is required.

Static reduction – a static web page reduction means having a reduced web page displayed without the ability of further browsing, without the animation of animated gif images, no video clips running, or any other moving objects (marquees, certain applets, etc.).

Dynamic reduction – a dynamic web page reduction means displaying a reduced web page, while maintaining the ability to interact with the site, further browse other pages by clicking on hyperlinks, and having the animation and movement of animated GIF images and other moving entities.

Static reduction

When static reduction is applied, we reduce the already grabbed (see the previous paragraph - the “grabbing phase”) image of a web page, by applying in house implemented reducing and image enhancement algorithms.

The reduction process takes into consideration the fact that a common web page is designed to be displayed on a target window characterized by a resolution of 600x800 pixels. Therefore, the transformation function of the reducing component transforms the web page image from a window of 600x800 pixels to a window which has the dimensions of the target WebSplit client’s frame, as it is configured at the time the conversion is occurred. Since we want to display as much information as possible - this must be done in order to avoid the need of appending scroll bars to the image window.

Displaying phase

A web page is displayed **46** after it has been reduced to be fit in its appropriate WebSplit frame. The displaying phase is simply about bringing the new “reduced” image to the frame window, while being able to refresh its content either on user demand or when an update event triggers the auto refreshing mechanism.

Caching system 42

The internal cache **47** is designed to allow WebSplit client application to speed up the rendering processes, and to bring the images to the appropriate frames without forcing the user to wait until the whole processing is done. The cache holds, among other things, two types of data:

bookmarked web pages – the actual “.html” files that were fetched from the appropriate web sites.

image files – that were captured from these “.HTML” files (see “grabbing phase” of the rendering engine paragraph) and represent the actual bitmap images (scale 1:1) of these bookmarked web pages.

When a bookmarked web page needs to be rendered to any of the split screen frames, the WebSplit client first queries the cache (as explained in the “auto update of a favorite bookmarked page” paragraph, of the WebSplit client application section), in order to avoid a too time-consuming process.

Auto update of a favorite bookmarked page

The WebSplit client application is always sensitive for favorite bookmarked pages updating events. By using an internal timer who wakes up an updating process every few minutes, it always looks for any changes done to any of its bookmarked pages.

In order to determine whether a change has been done to a bookmarked web page, and to avoid bringing the web page alone with its sub-components (images and such) every time (a procedure that may be time consuming) - the updating process first queries the internal cache. This query is carried out in order to find out whether the local cached version of the bookmarked web page and the one on the appropriate web site – are identical. If the two versions are not identical – an updating event is triggered.

When such an event is detected (meaning – a particular bookmarked page has altered), the client fetches the page again. Then, by re-employing the rendering engine on the new version of the bookmarked page, it refreshes the relevant frame content, and signals the user by either changing the surrounding frame color, or by masking the frame content with flashes of a different semi-transparent color.

Communities

Being able to communicate with WebSplit users is one of the key features of the product. Communities are groups of WebSplit users who are sharing common

ground since they either may have registered certain common web pages as their favorite bookmarks, or – at a certain moment, they are browsing the same content in the same web site. The WebSplit product allows bringing people with such mutual interest to a “rendezvous”.

Users lists

Each of the split screen frames of WebSplit client application may be operated in one of two modes: Content mode or Community mode. When on Content mode – the bookmarked web page is visible. When on Community mode – the split screen’s frame is occupied by a graphical representation of the users list, where users who have that page as one of their own bookmarks too, or actually surf to the same website at that moment - are displayed there.

Every user of the users list is represented by an icon who is designed to reveal the maximum possible information about the person behind it. Thus, the icons are designed as the two familiar sex symbols, to let the WebSplit user quickly find a person of a preferred gender to chat with, for instance. Moreover, the icons appear on the users list with different sizes – in order to imply better matching to the user personal preferences based on his selection of favorite bookmarks. Any of the icons in the users list is sensitive to a right mouse button click that raise a context menu, where communication functions (such as: send a message, chat with, ignore user, etc.) are available as well as the ability of viewing a possible counterpart’s personal details.

Communication among WebSplit users

Communication among WebSplit users is carried out by providing them with instant messaging and chatting utilities. WebSplit users who appear in the users list in a certain split screen frame can send or receive an instant message to any of the other users, or may participate in a chat. Users can be also set to be in an “ignore” state, to specify to the WebSplit client application to disregard instant messages or chatting requests coming from them.

Searching tools

Searching tools are applied by WebSplit users to find other people who fall into certain predefined criteria. By having the ability of managing a search profile and to set matching rules, people can find each other more efficiently. Among the criteria for searching other people there will be a search according to gender, age group, astrological sign, location, occupation, field of interest, etc.

WebSplit servers

The WebSplit server is a collection of software components that allow the user to manage his personal information along with his favorite bookmarks, acquire rating numbers calculated for him about web sites he is interested in, and communicate with his peers who may be connected with him - since they are watching the same content (surfing the same web pages or visiting the same web sites) or bookmark the same web sites as their favorites.

The WebSplit server has three main components:

WebSplit communication server – who only responsible for the communication between the various WebSplit clients with WebSplit database.

WebSplit database – is where all the various types of information is stored.

WebSplit server data managers – who are responsible for the statistical calculations, users list arrangements, etc.

Communication server

The communication server **50** is responsible for managing the communication between WebSplit clients with WebSplit databases. In order to achieve that, a communication abstraction layer **53** has been developed and incorporated into the WebSplit server and client – sharing the same API and methodology for both products (see “The communication model architecture” paragraph, of the “WebSplit server” section). All communication activities, data accessing, initiating communication sessions among WebSplit users, error handling, and others – are treated through and by the WebSplit communication server.

To avoid heavy load that may lead to poor performances – WebSplit clients connect to WebSplit server using a “connectionless” session paradigm. Connectionless session means connecting to the server only for login and authentication, and then disconnect – while maintaining an “online” status on WebSplit database, signaling to the server that the WebSplit client is still online, may be reached at any time (according to his IP address he registered with WebSplit server when he first issued the initial connection request), and should be treated as online client – when he renew the connection.

An internal mechanism checks every few minutes whether such an “online” client (who is not constantly connected) is still responding or issuing a request from time to time. Clients who lost connection and cannot be reached after considerable amount of time – are removed from the “online” users list, and need to log in again when they need to access the server.

WebSplit database

The WebSplit database **52** is the data warehouse of all the offline and online information. Here's where clients store their personal details and favorite bookmarks. Based on the personal information and searching profile - clients can run searching procedures to find their peers according to their mutual interest, predefined characteristics or qualities, etc. Users may find each other and communicate – since the database supply the infrastructure for them to meet with each other on the Internet.

WebSplit server data managers

Along with the on going activity of the WebSplit server, meaning – managing communication with various WebSplit clients and allowing accessing the WebSplit database for the various needed operation on the data it holds – the WebSplit server run separate procedures to manage the data and to deduct new types of information that may be valuable. For such purposes, statistical data is calculated out of the current information in WebSplit databases – regarding users favorite web sites ratings and hot spots ranking.

Domains rating manager

The domain rating manager **54** is responsible of rating the huge collection of the various web sites who are bookmarked by WebSplit users in the WebSplit database. According to data supplied from the various online WebSplit clients, it keeps track and calculates for each and every one of the bookmarked web sites the following numbers:

How many online users surf the web site at a given time, using their browser.

How many online user already bookmarked this web site.

How many user (both online and offline) already bookmarked this web site.

These numbers appear on the WebSplit client application on the corresponding split screen frames, so WebSplit users can see them when they connect to the WebSplit server, or when the WebSplit clients they run are triggered by updating events.

Hot spots ranking manager

The hot spots ranking manager **55** is responsible for ranking the top and most popular bookmarked web sites. The process runs on the list of all bookmarked web pages who are already registered in the WebSplit database, and keep them sorted according to their (changing and maybe “fluctuating”) popularity. In other words, web sites that are popular and are registered as favorite bookmarks by many users – will be positioned higher in the ordered list. On the other hand – web sites that are less popular; and not many users register them as their favorite bookmarks – will be probably positioned closer to the bottom of the list.

WebSplit user profiles manager

While the user only keeps on WebSplit servers his own personal favorite bookmarks along with some minimal personal details, it allows us to deduct valuable information out of it. The user’s favorite bookmarks are checked to find some important meta-tag’s keywords that may imply about his personal preferences.

The communication model architecture

The communication model 60 allows WebSplit clients 61 and server 62 to communicate over the internet 63, let every user, using any computer anywhere in the world – have access to WebSplit databases, view on the split screen window his favorite bookmarks, always updated to last moment, and to find online peers with common interests who associated with his favorite sites.

The following points are the most important ideas behind the design and implementation of the communication layer infrastructure:

- Simple API for communication management purposes.
- Same API for both the WebSplit client and the communication server.
- Handling multi concurrently running sessions at the same time.
- Implementation using traditional sockets programming model.
- Two layers of abstraction: communication management, and protocols.

The communication management layer

The communication management layer is an abstraction of all the necessary infrastructure that needs to expose a reasonable communication management API to both WebSplit communication server and client applications.

The protocols layer

The protocol layer 66 incorporates all the infrastructure that hides the internal WebSplit protocols 67, 68 for the various services who involve any thing to do with communication. These internal protocols who actually define all possible “handshakings” and “dialogs” between WebSplit client 61 and communication server, are represented in their low level form by a finite set of packets who travel back and forth.

For the definition to be excluded from the WebSplit client and communication server 62 applications, and to make a better and more accurate and bug free

employment – we designed an API who free these applications from handling the complexity of the communication packets constructing and deconstructing.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

WebSplit Imaging concept: WebSplit's preferred embodiment application is built from several components, which together fit into a unique solution of which allows users to view several web pages simultaneously. Integral to this concept, it can be stated without loss of generality that the present invention relates to a rendering engine, tangibly embodying a program of instructions executable by the machine to perform method steps 70 for presenting split screen bookmark imagery in conjunction with a fetch and convert engine and with a repository. Accordingly, hereinafter is a description of the main components (see figure 7) including: Repository 71, Fetch and Convert engine 72, Rendering engine 73.

A typical operational scenario would be:

1. Rendering engine request for new content
2. Fetch and Repository engine check the repository for update content
3. If update content exists, Fetch and Convert engines hands the updated content to the Rendering engine
4. Otherwise Fetch and Convert engine download and filter new viewable content
5. On complete download, the Fetch and Convert engines converts the content to image
6. New image is saved using the repository while cropping and compressing
7. Rendering engine notifies when content is ready
8. New image is being displayed and filtered

1. Rendering engine:

The rendering engine display several images simultaneously, each image is filtered with image enhancement filter; the actual code, which displays the image, will not be covered here.

The following is an excerpt from the filter function:

1.1 First - creating an intermediate image to hold horizontal zoom

```
fwidth = workImage.GetFilterWidth() ;
UCHAR *tmp = new UCHAR{DstWidth * SrcHeight * 3};
xscale = (double) (DstWidth-1) / (double) (SrcWidth-1);
yscale = (double) (DstHeight-1) / (double) (SrcHeight-1);
```

1.2 Next - pre-calculating filter contributions for a row,

```
contrib{i}.n = 0;
contrib{i}.p = (CONTRIB *)calloc((int) (fwidth * 2 + 1),
sizeof(CONTRIB));
if (i>maxAlloc)
    maxAlloc = i;
center = (double) i / xscale;
left = ceil(center - fwidth);
right = floor(center + fwidth);
for(j = left; j <= right; ++j)
{
    weight = center - (double) j;
    weight = workImage.Filter(weight);
    if(j < 0)
    {
        n = -j;
    }
    else if (j >= SrcWidth)
    {
        n = (SrcWidth - j) + SrcWidth - 1;
```

```

    }
    else
    {
        n = j;
    }
    k = contrib{i}.n++;
    contrib{i}.p{k}.pixel = n;
    contrib{i}.p{k}.weight = weight * 65536.0;
}

```

1.3 Finally – placing each calculated pixel into a common intermediate image

```

r = 0;
g = 0;
b = 0;
for(j = 0; j < contrib{i}.n; j++)
{
    xweight = contrib{i}.p{j}.weight;
    offset = (k * orgBytesPerLine) + (contrib{i}.p{j}.pixel*3);
    bitsOffset = (UCHAR *)workImage.m_ppvBits;
    r += bitsOffset[offset + 0] * xweight;
    g += bitsOffset[offset + 1] * xweight;
    b += bitsOffset[offset + 2] * xweight;
}
r = (r >> 16);
g = (g >> 16);
b = (b >> 16);
if (r<0) r=0;
if (g<0) g=0;
if (b<0) b=0;
if (r>255) r=255;
if (g>255) g=255;

```

```

    if (b>255) b=255;
    offset = (i+k*DstWidth)*3;
    tmp{offset+0} = (UCHAR)r;
    tmp{offset+1} = (UCHAR)g;
    tmp{offset+2} = (UCHAR)b;
}

```

2. Fetch and Convert engine:

The Fetch and Convert engine is the core of WebSplit solution, its main responsibilities are:

2.1. Fetching viewable content from the web excluding/filtering none viewable content (script, ActiveX controls etc....)

2.2. Convert engine: This part of the component convert the viewable content to image and pass to the repository

following is and Excerpt from the Fetch engine source code:

2.2.1 This is the main entry point in which the rendering image engine request viewable content from the Fetch and Convert Engine, on upon entry to this function the Fetch and Convert engine

check the repository for an update image if none update image exist the component

turn onto the web in order to retrieve a new and update content

```

LoadDocumentHtml(navigateUrl,
countID,Url,,LOADPROPERTIES,eLoadProperties )

```

```

{

```

```

    HRESULT vr_hRes;

```

```

    :

```

```

    switch( vr_eLoadProperties )

```

```

    {

```

```

        case downloadNormal_ :

```

```

        if(( vr_hRes = VerificationCache( vr_countID, vr_Url )) == S_OK )

```

```

            return vr_hRes;

```

```

        else

```



```

        break;
    case downloadRefresh_ :
        break;
    }
    DEFINE_TRANSACTION obj_defTransaction;
    :
    :
    :
    }

```

2.2.2 GetThumb - uses Activex control internal capabilities in order to render it's content on a memory surface

```

_GetThumb( HBITMAP& vr_hThumb )
{
    RECTL rcBounds = { 0, 0, vr_lWidth, vr_lHeight };
    vr_hRes = pnt_ViewObject->Draw( DVASPECT_CONTENT,
                                    -1,
                                    NULL,
                                    NULL,
                                    0,
                                    hdcMem,
                                    &rcBounds,
                                    NULL,
                                    NULL,
                                    0 );
    :
    :
    :
}

```

2.2.3 Next the memory surface is convert into image and saved in the repository

```

HRESULT _SaveThumb( obj_ThumbNail, obj_Url, ,obj_Path)
{

```

```

:
:
    pnt_CacheManager->PutImage( pnt_PictureDisp, obj_Url, &obj_Path
)))
    {
:
:
    }
}

```

3.Repository

The repository contain a collection of data structures which represent viewable content

Each data structure hold the actual image for the viewable content and several properties:

Last update time, source URL and more...

following is and Excerpt from the Repository source code:

3.1 Adding new entry to the repository as follow

PutImage(IPictureDisp *pPictureDisp,BSTR Url,BSTR *Path)

```

{
    hRes = pPictureDisp>Invoke(.....
    CDib dibBitmap;
    HBITMAP hBitmap = (HBITMAP)varResult.lVal;
    dibBitmap.SetBitmap(hBitmap);
    dibBitmap.Crop(CRect(3,3,dibBitmap.GetWidth()-3,dibBitmap.GetHeight()-
3),TRUE);

```

```

CCacheEntry newEntry;
newEntry.m_szUrl = csUrl;
newEntry.m_szImagePath=csPath;
newEntry.m_dtCreationDate= _GetSystemTime();
newEntry.m_dtLastViewDate = _GetSystemTime();

```

```
newEntry.m_dwFileSize = _GetFileSize(csPath);  
:  
:  
BOOL bRC = m_CacheEnteries.Add(newEntry);  
}
```

3.2 Retrieve entry from the repository

GetImage(BSTR Url, long* needUpdate, BSTR *Path)

```
{  
:  
:  
:  
time_t vTime;  
    :: time (&vTime);  
    if(vTime - mCreationDate > m_RefreshInterval * WS_SECOND)  
        *needUpdate = 1;  
    else  
        *needUpdate = 0;  
    CComBSTR bsPath;  
    bsPath=csPath.AllocSysString();  
    bsPath.CopyTo(Path);  
}
```

CLAIMS

1. A facile method of forming a dynamic psycho-ergonomic profile of users, for correlation of like users, and accordingly for forming common profile "societies", the method including the steps of:
 - a) **forming** a repository for personal bookmark lists wherein the repository is accessible to at least one network server;
 - b) each user of a plurality of users **maintaining** his respective personal bookmark list in the repository;
 - c) logically **aggregating** a collective list of all bookmarks maintained in the repository wherein
 - i) each bookmark is listed only once, and
 - ii) each user maintaining that respective bookmark is listed in association therewith;
 - d) logically **structuring** a combinatorial set of n-tuplings of bookmarks in the collective list wherein for each n-tupling there are logical links with each respective user maintaining every bookmark in the n-tupling; and
 - e) **providing** substantially each user of the plurality of user with read-only access to the logical links of any n-tupling wherein said each user is likewise logically linked.
2. The method according to claim 1 wherein said step of **providing** includes **installing** a rendering engine on a terminal associated process of a user of any user of the plurality of users, wherein the rendering engine maps content associated with the logical links of read-only access onto the associated terminal.
3. The method according to claim 1 wherein said step of **maintaining** includes **accepting** personal data about the respective user.

4. The method according to claim 3 wherein said step of **accepting** includes **querying** the respective user for his personal data, **receiving** his response to the query, and **associating** a stored representation of the response with his respective personal bookmark list.
5. The method according to claim 4 wherein said step of **aggregating** includes **characterizing** the stored representation of a respective user's personal data response and **maintaining** this characterization as at least one pseudo-bookmark in the user's respective personal bookmark list, wherein pseudo-bookmarks are aggregated into the collective list substantially like other maintained bookmarks.
6. The method according to claim 3 wherein said step of **accepting** includes **querying** a database for personal data associated with the respective user, **receiving** a response to the query, and **associating** a stored representation of the response with the user's respective personal bookmark list.
7. The method according to claim 6 wherein said step of **aggregating** includes **characterizing** the stored representation of a respective user's data-base provided personal data and **maintaining** this characterization as at least one pseudo-bookmark in the user's respective personal bookmark list, wherein pseudo-bookmarks are aggregated into the collective list substantially like other maintained bookmarks.
8. A rendering engine, tangibly embodying a program of instructions executable by the machine to perform method steps for presenting split screen bookmark imagery in conjunction with a fetch and convert engine and with a repository, the method including the steps of
 - a) **creating** an intermediate image to hold horizontal zoom;
 - b) **pre-calculating** filter contributions for a row; and
 - c) **placing** each calculated pixel into a common intermediate image.

9. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for forming a dynamic psycho-ergonomic profile of users, said method steps including:

- a) **forming** a repository for personal bookmark lists wherein the repository is accessible to at least one network server;
- b) each user of a plurality of users **maintaining** his respective personal bookmark list in the repository;
- c) logically **aggregating** a collective list of all bookmarks maintained in the repository wherein
 - i) each bookmark is listed only once, and
 - ii) each user maintaining that respective bookmark is listed in association therewith;
- d) logically **structuring** a combinatorial set of n-tuplings of bookmarks in the collective list wherein for each n-tupling there are logical links with each respective user maintaining every bookmark in the n-tupling; and
- e) **providing** substantially each user of the plurality of user with read-only access to the logical links of any n-tupling wherein said each user is likewise logically linked.

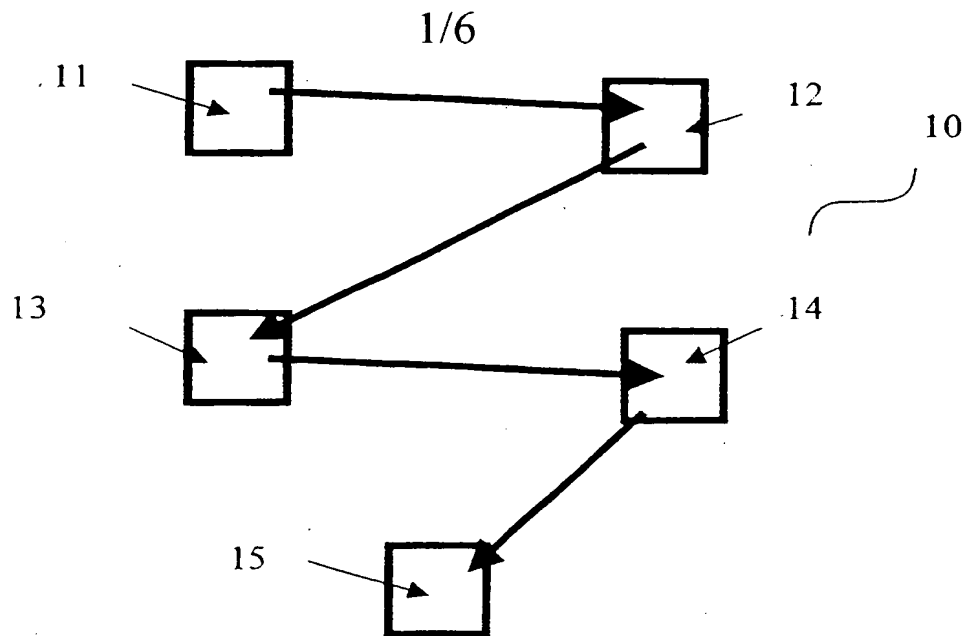


Figure 1

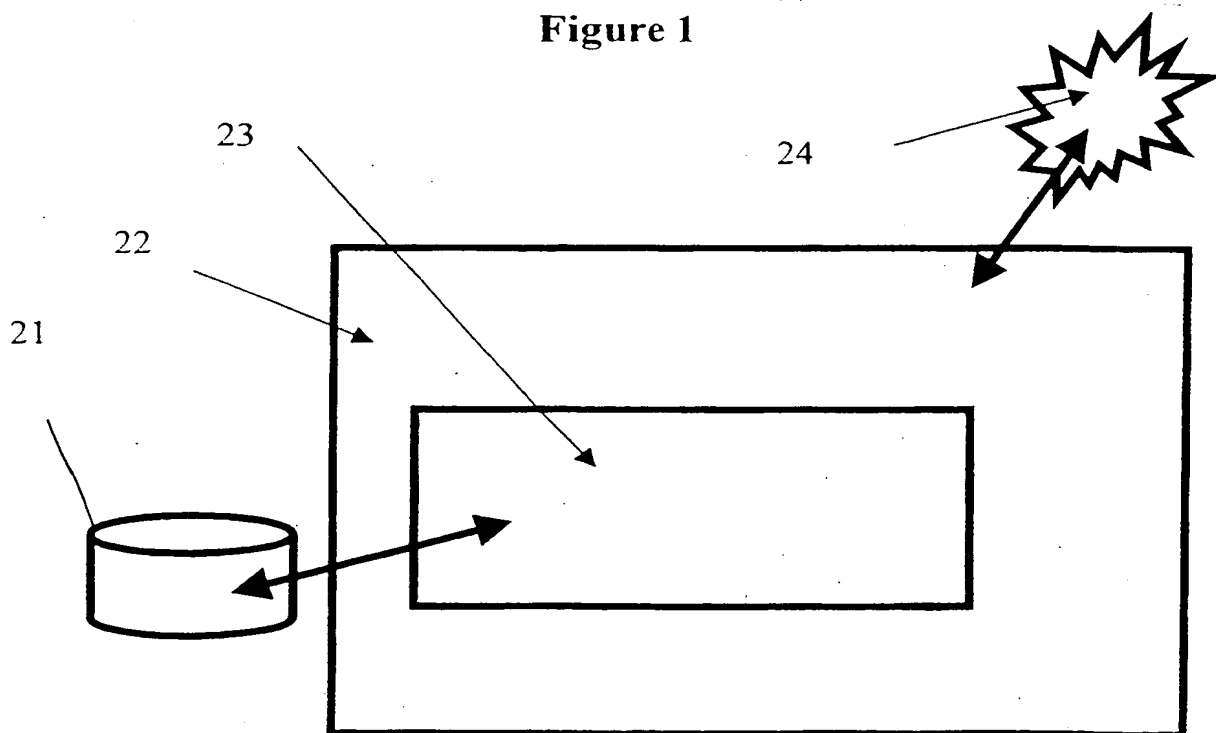


Figure 2

2/6

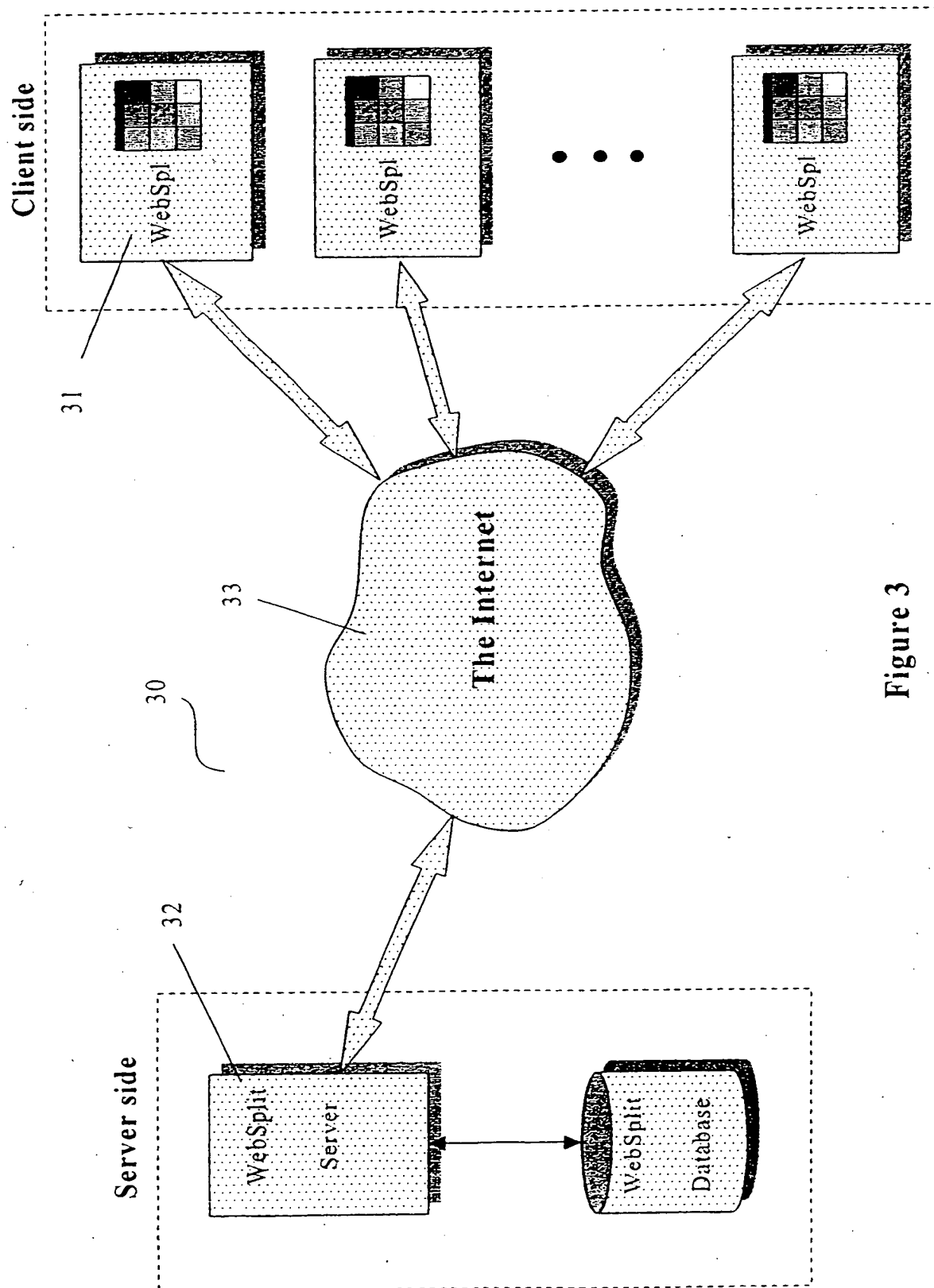


Figure 3

WebSplit - the client-server

3/6

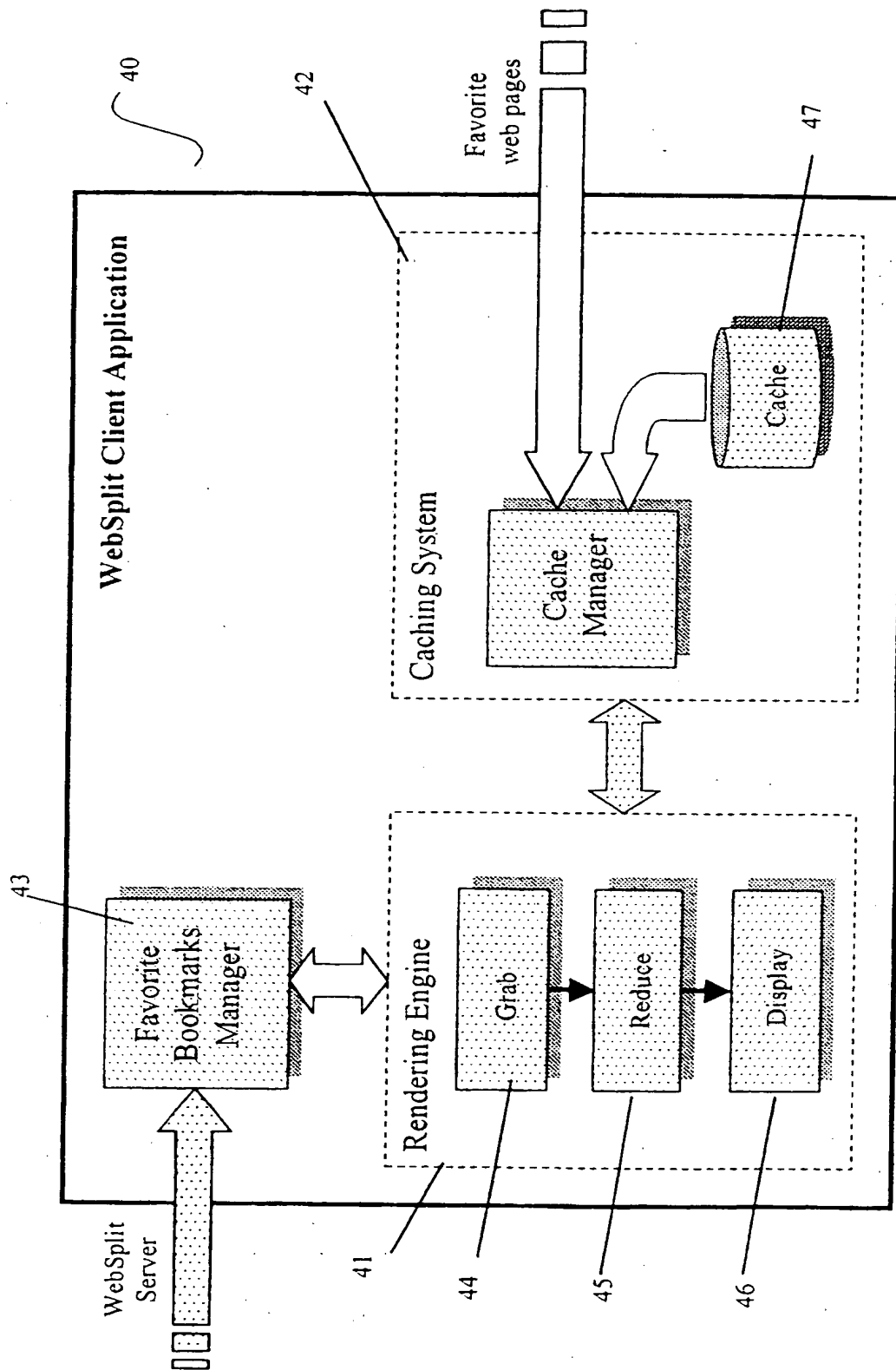


Figure 4

WebSplit – favorite bookmarks management and display

4/6

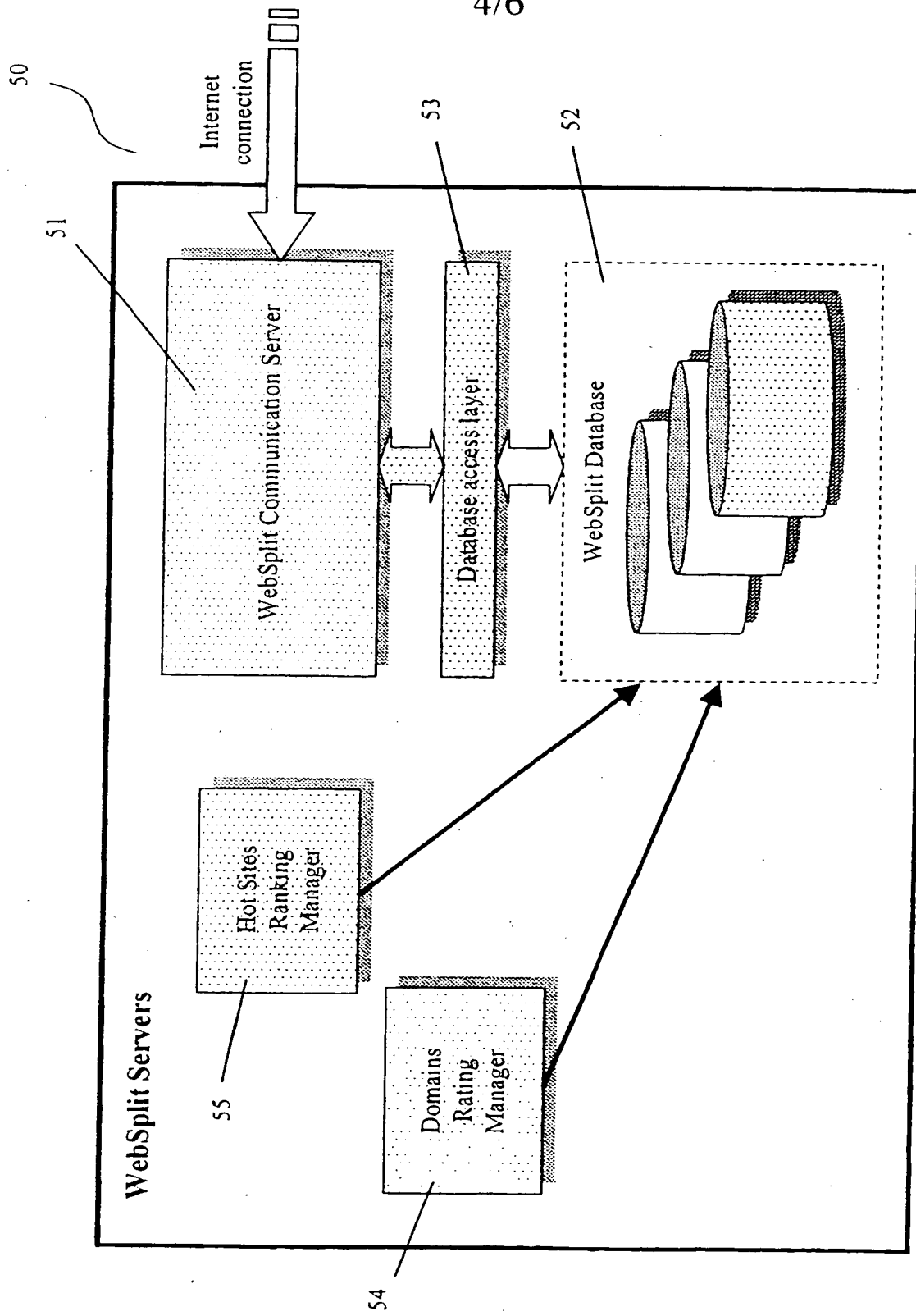


Figure 5
WebSplit – servers architecture

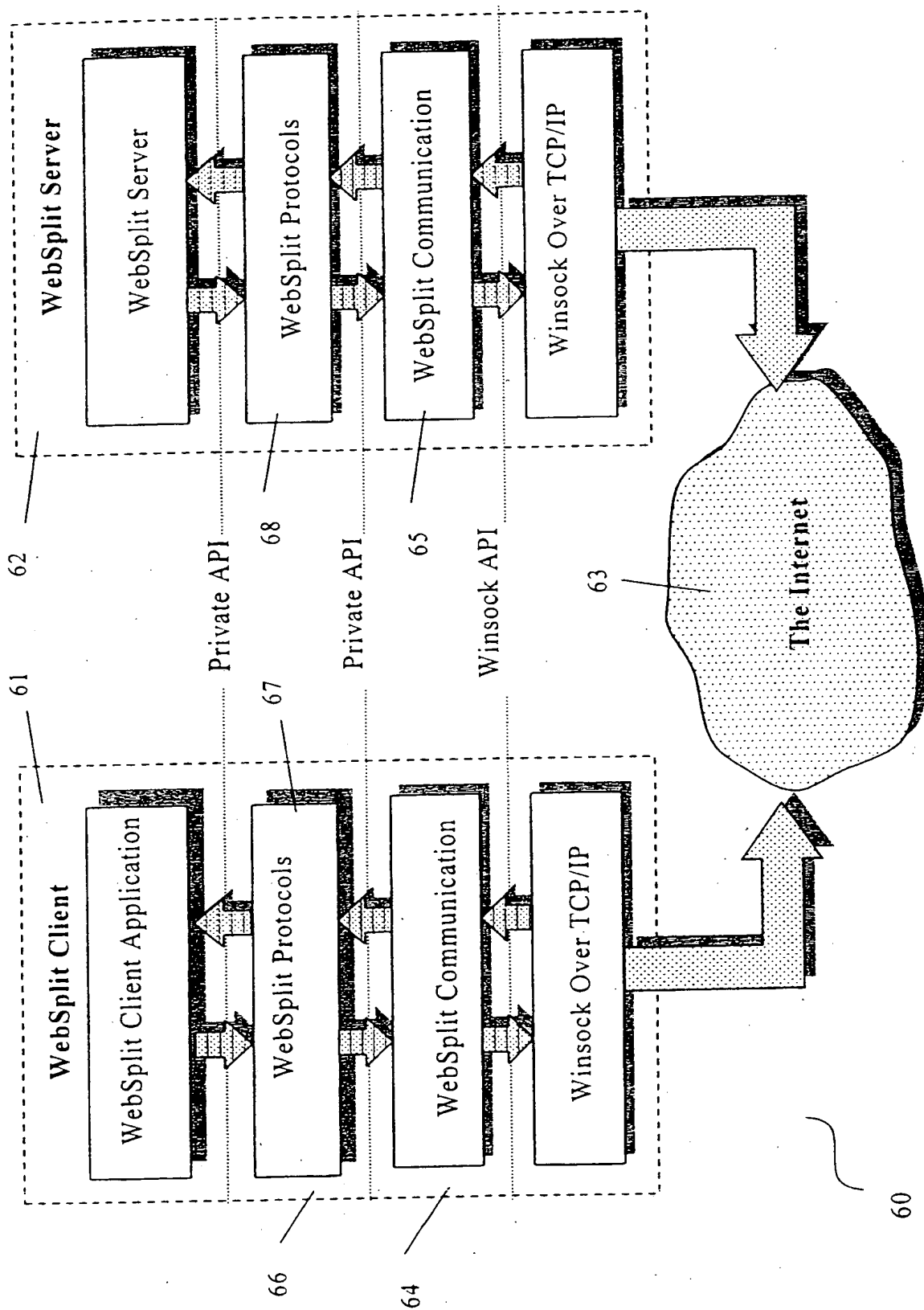


Figure 6
WebSplit - communication model

6/6

70

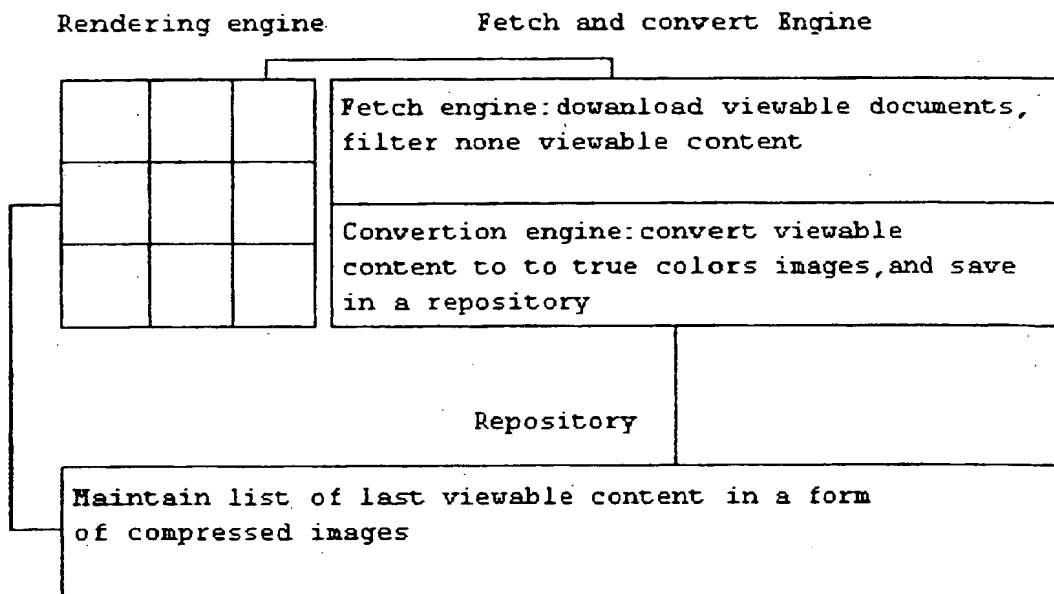


Figure 7

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/24508 A3

(51) International Patent Classification⁷: **G06F 17/30**

(21) International Application Number: **PCT/IL00/00607**

(22) International Filing Date:
28 September 2000 (28.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
132144 30 September 1999 (30.09.1999) IL

(71) Applicant (for all designated States except US): **WEB-SPLIT.COM INC.** [US/US]: 1209 Orange Street, County of Newcastle, Wilmington, DE 19801 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **HAROUSH, Ariel** [IL/IL]; 18 Habaron Hirsch Street, 96148 Jerusalem (IL).

(74) Agent: **JEREMY M. BEN-DAVID & CO. LTD.**; Har Hotzvim Hi-Tech Park, P.O. Box 45087, 91450 Jerusalem (IL).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

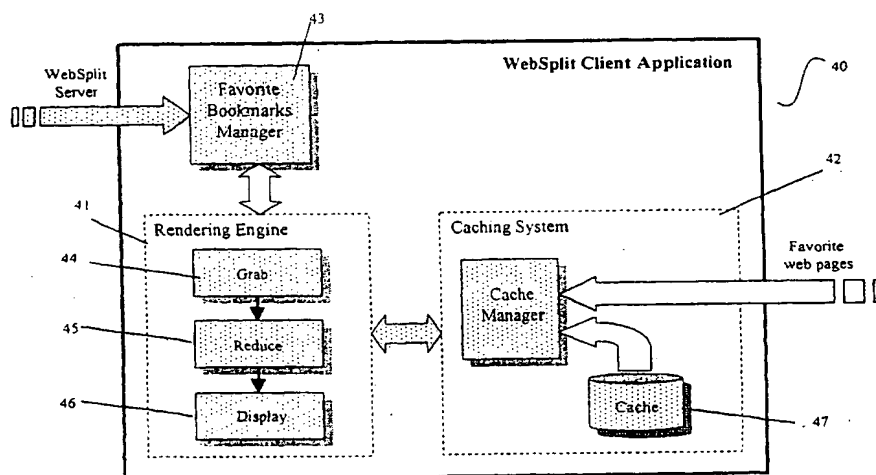
Published:

— with international search report

(88) Date of publication of the international search report:
18 October 2001

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD OF FORMING NETWORK USER PROFILES



WebSplit – favorite bookmarks management and display

(57) Abstract: Disclosed is a method of forming a profile of users for correlation of like users, the steps include: forming a repository for personal bookmark lists (43); maintaining respective personal bookmark list in the repository; logically aggregating a collective list of all bookmarks maintained in the repository; logically structuring a set of n-tuplings of bookmarks in the collective list; and providing each user with read-only access to the logical links of any n-tupling wherein said each user is logically linked. Furthermore, disclosed is a rendering engine (41), embodying a program of instructions executable by the machine to perform method steps for presenting split screen bookmark imagery in conjunction with a convert engine and with the repository, the method including the steps of creating an intermediate image to hold horizontal zoom; pre-calculating filter contributions for a row; and placing each calculated pixel into a common intermediate image.

INTERNATIONAL SEARCH REPORT

 International application No.
PCT/IL00/00607

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/30

US CL : 707/10, 104., 513; 709/201,203

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/10, 104., 513; 709/201,203

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

 Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,895,471 A (KING et al) 20 April 1999, abstract, col.11 line 31 through col.12 line 30	1-9
Y	US 5,918,010 A (APPLEMAN et al) 29 June 1999, abstract, figure 9, figure 13, 13B and 13C	1-8
A	US 5,813,007 A (NIELSEN) 22 September 1998, abstract, figure 2 and figure 4	1-9
Y.P	US 6,108,645 A (EICHSTAEDT et al) 22 August 2000, abstract, figure 3, col.8 lines 48-64, col.9 1-35, col.10 lines 55-64	1-9

☐ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y"	document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means		
"P" document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

02 FEBRUARY 2001

Date of mailing of the international search report

05 APR 2001

 Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

GRETA L. ROBINSON

Telephone No. (703) 308-7565